Amendments to the Claims:

This listing of claims will replace all prior versions and listings of claims in the application:

Listing of Claims:

1. (Currently Amended) A system for providing a floating point product, comprising:

an analyzer circuit configured to determine a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

a results circuit coupled to the analyzer circuit and configured to assert a resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand, wherein each of the first floating point operand, second floating point operand and resulting floating point operand comprises a sign bit, an exponent field and a fraction field, and wherein at least one of the five lowest order bits of the fraction field of one of the first floating point operand, second floating point operand and resulting floating point operand comprises at least one flag.

2. (Original) The system for providing a floating point product of claim 1, wherein the analyzer circuit further comprises:

a first operand buffer configured to store the first floating point operand;

a second operand buffer configured to store the second floating point operand;

a first operand analysis circuit coupled to the first operand buffer, the first operand analysis circuit configured to generate a first characteristic signal having information relating to the first status; and

a second operand analysis circuit coupled to the second operand buffer, the second operand analysis circuit configured to generate a second characteristic signal having information relating to the second status.

3. (Original) The system for providing a floating point product of claim 2, wherein the first status and the second status are determined without regard to memory storage

external to the first operand buffer and the second operand buffer.

4. (Original) The system for providing a floating point product of claim 3, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

5. (Original) The system for providing a floating point product of claim 1, wherein the results circuit further comprises:

a multiplier circuit coupled to the analyzer circuit, the multiplier circuit configured to produce the product of the first floating point operand and the second floating point operand;

a multiplier logic circuit coupled to the analyzer circuit and configured to produce the resulting status based upon the first status and the second status; and

a result assembler coupled to the multiplier circuit and the multiplier logic circuit, the result assembler configured to assert the resulting floating point operand and embed the resulting status within the resulting floating point operand.

6. (Previously Presented) The system for providing a floating point product of claim 5, wherein the multiplier logic circuit is organized according to a structure of a decision table.

7. (Previously Presented) The system for providing a floating point product of claim 1, wherein the product of the first floating point operand and the second floating point operand is identical in all cases to the product that would be produced if the two operands were swapped.

8. (Original) The system for providing a floating point product of claim 1, wherein the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

9. (Previously Presented) The system for providing a floating point product of claim 8, wherein the overflow status represents one in a group of a plus overflow (+OV) status and a minus overflow (-OV) status.

10. (Original) The system for providing a floating point product of claim 8, wherein the overflow status is represented as a predetermined non-infinity numerical value.

11. (Previously Presented) The system for providing a floating point product of claim 8, wherein the underflow status represents one in a group of a plus underflow (+UN) status and a minus underflow (-UN) status.

12. (Original) The system for providing a floating point product of claim 8, wherein the underflow status is represented as a predetermined non-zero numerical value.

13. (Original) The system for providing a floating point product of claim 8, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

14. (Original) The system for providing a floating point product of claim 8, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

15. (Currently Amended) A method for providing a floating point product, comprising:

determining a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; and

asserting a resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand, wherein each of the first floating point operand, second floating point operand and resulting floating point operand comprises a sign bit, an exponent field and a fraction field, and wherein at least some of the next lowest order bits after the five lowest order bits of the fraction field encode

additional information in relation to the first, second and resulting statuses.

16. (Original) The method for providing a floating point product of claim 15, wherein the determining stage further comprises:

storing the first floating point operand in a first operand buffer;

storing the second floating point operand in a second operand buffer;

generating a first characteristic signal representative of the first status; and

generating a second characteristic signal representative of the second status.

17. (Original) The method for providing a floating point product of claim 16, wherein the first characteristic signal and the second characteristic signal are generated without regard to memory storage external to the first operand buffer and the second operand buffer.

18. (Original) The method for providing a floating point product of claim 17, wherein the memory storage external to the first operand buffer and the second operand buffer is a floating point status register.

19. (Original) The method for providing a floating point product of claim 15, wherein the asserting stage further comprises:

producing the product of the first floating point operand and the second floating point operand; and

asserting the resulting floating point operand having the resulting status embedded within the resulting floating point operand.

20. (Original) The method for providing a floating point product of claim 15, wherein the product of the first floating point operand and the second floating point operand is identical in all cases to the product that would be produced if the two operands were first swapped.

21. (Original) The method for providing a floating point product of claim 15, wherein

the first status, the second status, and the resulting status are each one of the following: an invalid operation status, an overflow status, an underflow status, a division by zero status, an infinity status, and an inexact status.

22. (Previously Presented) The method for providing a floating point product of claim 21, wherein the overflow status represents one in a group of a plus overflow (+OV) status and a minus overflow (-OV) status.

23. (Original) The method for providing a floating point product of claim 22, wherein the overflow status is represented as a predetermined non-infinity numerical value.

24. (Previously Presented) The method for providing a floating point product of claim 21, wherein the underflow status represents one in a group of a plus underflow (+UN) status and a minus underflow (-UN) status.

25. (Original) The method for providing a floating point product of claim 24, wherein the underflow status is represented as a predetermined non-zero numerical value.

26. (Original) The method for providing a floating point product of claim 21, wherein the invalid status represents a not-a-number (NaN) status due to an invalid operation.

27. (Original) The method for providing a floating point product of claim 21, wherein the infinity status represents one in a group of a positive infinity status and a negative infinity status.

28-40. (Canceled)

41. (New) The system for providing a floating point product of claim 1, wherein the at least one flag comprises at least one selected from the group of an invalid operation flag "n", an overflow flag "o", an underflow flag "u", a division-by-zero flag "z", and an inexact flag "x".

42. (New) The system for providing a floating point product of claim 1, wherein the next five lowest order bits after the five lowest order bits of the fraction field of the one of the first floating point operand, second floating point operand and the resulting floating point operand comprises at least one binary zero and at least one binary one.

43. (New) The method for providing a floating point product of claim 15, wherein the additional information is in relation to an operation and/or types of operands giving rise to a not a number ("NaN") status.

44. (New) The method for providing a floating point product of claim 15, wherein the at least some of the next lowest order bits after the five lowest order bits of the fraction field comprises the next five lowest order bits after the five lowest order bits of the fraction field.

45. (New) The method for providing a floating point product of claim 15, wherein the next five lowest order bits after the five lowest order bits of the fraction field comprise at least one binary zero and at least one binary one.

46. (New) A system for providing a floating point product, comprising:

an analyzer circuit configured to determine a first status of a first floating point operand and a second status of a second floating point operand based upon data within the first floating point operand and data within the second floating point operand respectively; wherein the analyzer circuit comprises:

a first group of comparators that generate asserted signals responsive to bits in an exponent field of the first and second floating point operands;

a second group of comparators that generate asserted signals responsive to bits of a first portion of a fraction field of the first and second floating point operands; and

a third group of comparators that generate asserted signals responsive to bits of a second portion of the fraction field of the first and second floating point

operands; and

a results circuit coupled to the analyzer circuit and configured to assert a resulting floating point operand containing the product of the first floating point operand and the second floating point operand and a resulting status embedded within the resulting floating point operand.

47. (New) The system for providing a floating point product of claim 46, wherein the bits of the first portion of the fraction field are of a higher order than the bits of the second portion of the fraction field.

48. (New) The system for providing a floating point product of claim 47, wherein the bits of the second portion of the fraction field comprise the five lowest order bits of the fraction field.

49. (New) The system for providing a floating point product of claim 48, wherein at least one of the bits of the second portion of the fraction field comprises a flag.

50. (New) The system for providing a floating point product of claim 49, wherein the flag comprises at least one selected from the group of an invalid operation flag "n", an overflow flag "o", an underflow flag "u", a division-by-zero flag "z", and an inexact flag "x".

51. (New) The system for providing a floating point product of claim 49, wherein the next five lowest order bits after the five lowest order bits of the second portion of the fraction field encode additional information in relation to the first, second and resulting statuses.

52. (New) The system for providing a floating point product of claim 51, wherein the additional information is in relation to an operation and/or types of operands giving rise to a not a number ("NaN") status.

53. (New) The system of claim 46, wherein each of the first and second portions of the fraction field comprises all binary ones.